iPhone Performance Optimization with Instruments

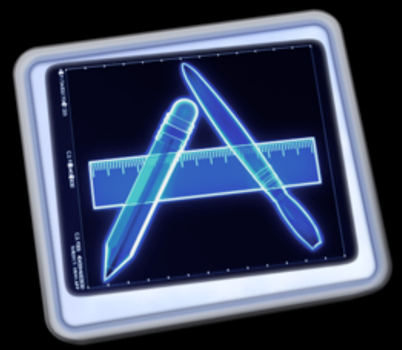
Session 414

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

iPhone Performance Optimization with Instruments

Daniel Delwood
Performance Tools Engineer

Performance



Proactive and Reactive

Use Instruments to identify and solve iPhone app performance problems

What You'll Learn

- Simulator vs. device
- Instrumentation for common iPhone issues
- Reacting to crashes/hangs/termination

Simulator and Device

What's the difference?

The Simulator

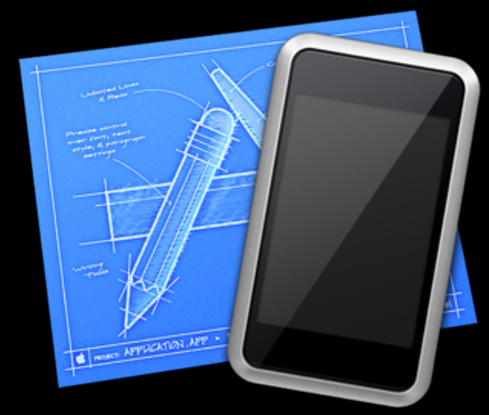
What is it

- API simulation
  - Performance of a Mac
  - Behaves like an iPhone
- Runtime environment
  - Apps installed

Profiling on the Simulator

When to use

- Memory usage
- File access



Behavior and Memory

Profiling on iPhone

Real performance

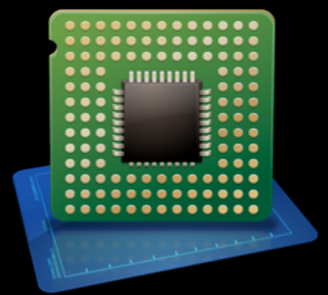
- Accurate timing
- ARM processor
- Limited resources
- User experience

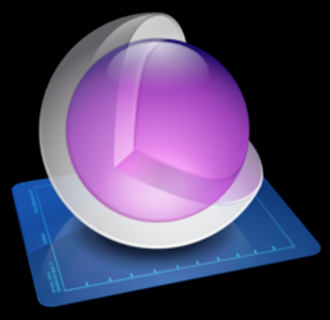
vs

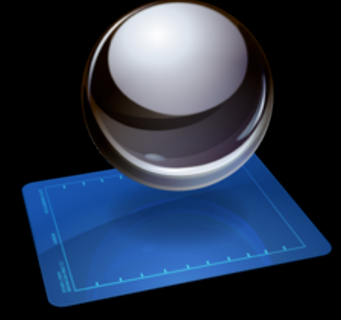
Speed and Responsiveness

Profiling on a Device

Available instrumentation



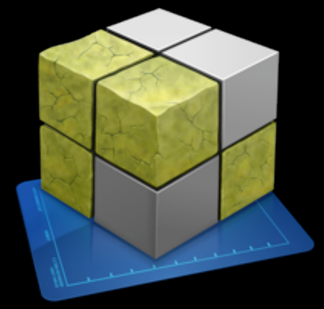




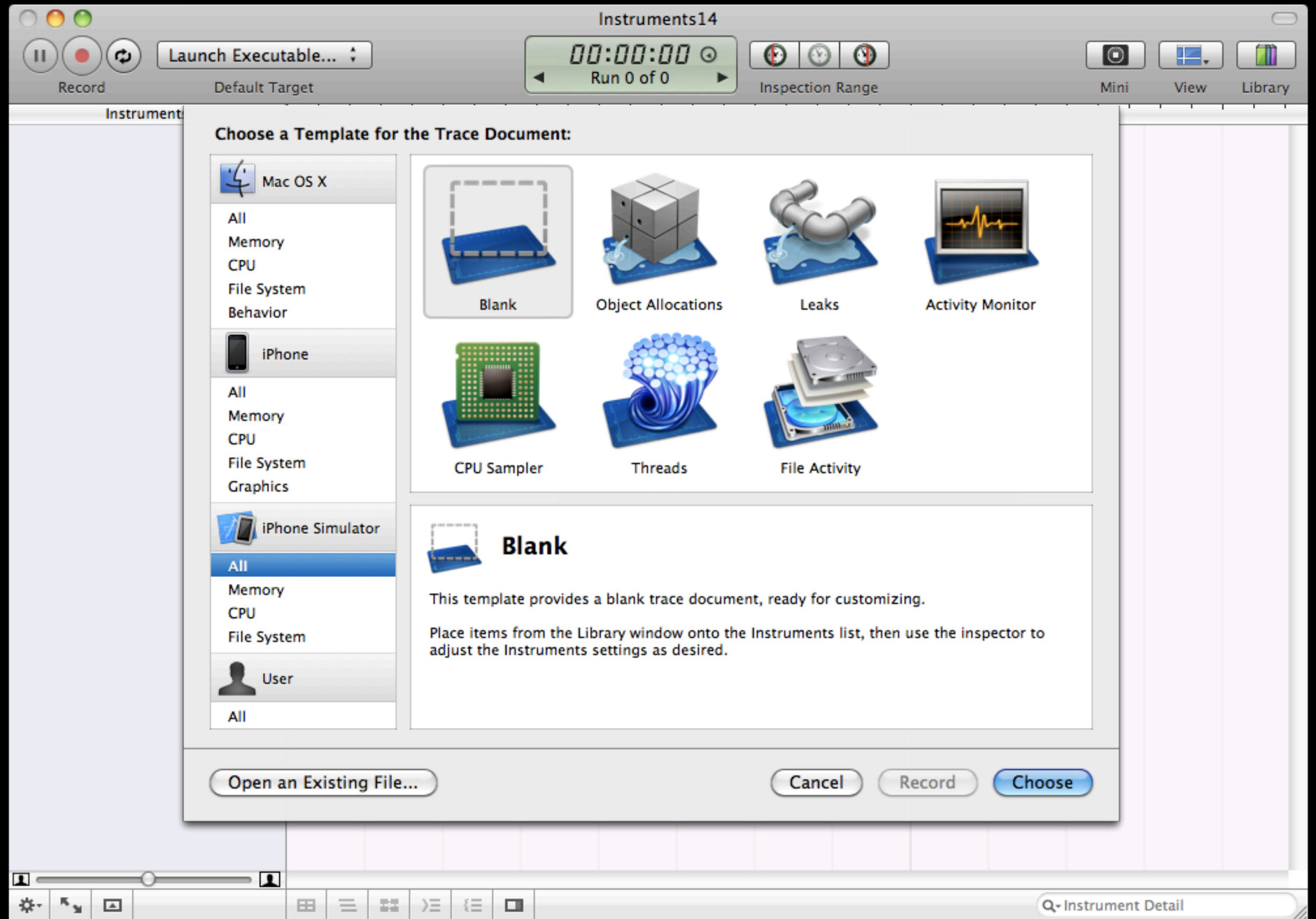
Profiling in the Simulator

Available instrumentation

- More than the device
  - DTrace
  - Time Profiler
  - Zombies



Instruments with SDK



Optimizing for Times

Launch faster, do less

Launch Times Demo

Lynne Salameh
Performance Tools Engineer

Time Tools

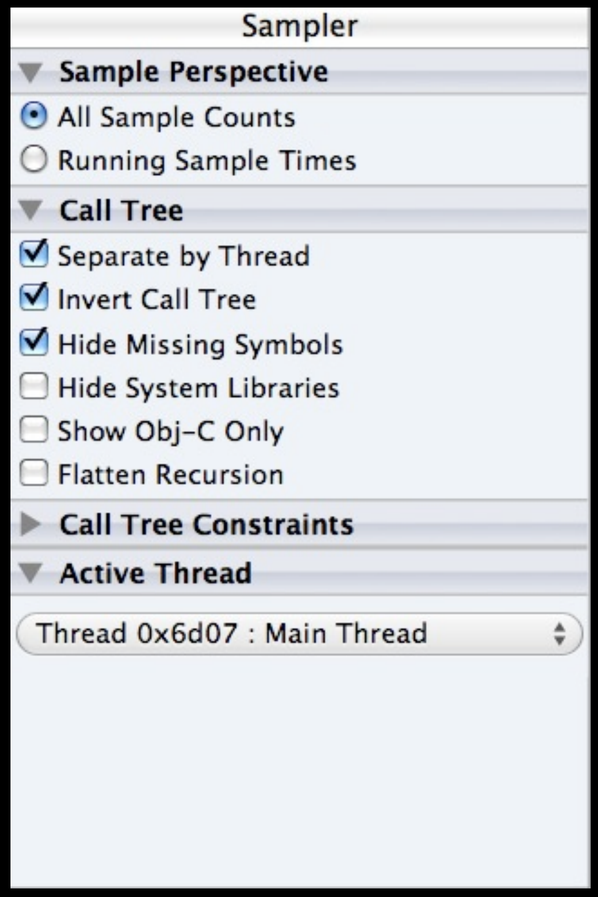
Time Tools

Sampler

- Backtraces/time-interval
- All thread states
- Statistical, not function profiling
- Find hotspots in your code

Statistical Sampling

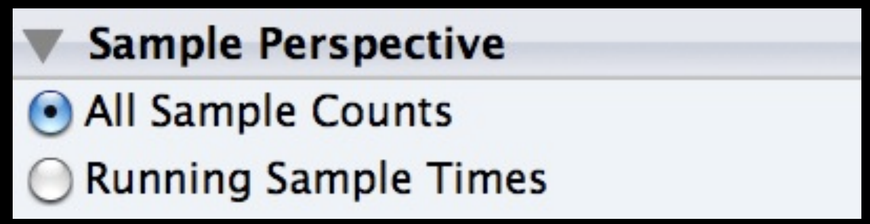
More details



Statistical Sampling

More details

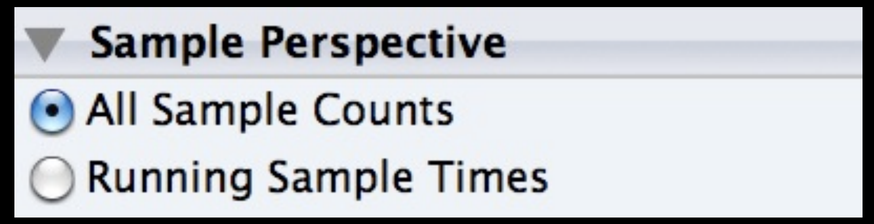
- Perspective
  - Counts → Thread States
  - Times → Executing threads



Blocked vs. Running

Statistical Sampling

More details



Samples	Library	Symbol
151		▼ Thread 0x5203 : Main Thread
151	Breadcrumbs	▼ main
151	UIKit	▼ UIApplicationsMain
151	GraphicsServices	▼ GSEventRunModal
102	libSystem.B.dylib	mach_msg_trap
26	Breadcrumbs	▼ -[RootViewController tableView:cellForRowAtIndexPath:]
11	Breadcrumbs	► -[BreadcrumbCell initWithFrame:reuseIdentifier:]
9	Breadcrumbs	► -[BreadcrumbEntry thumbnail]
6	Breadcrumbs	► -[RootViewController entryForIndexPath:]
8	libobjc.A.dylib	objc_msgSend
2	WebCore	► -[NSString(WebStringDrawing) _web_sizeWithFont:]
2	CoreGraphics	► CGContextDrawImage
...	...	...

Statistical Sampling

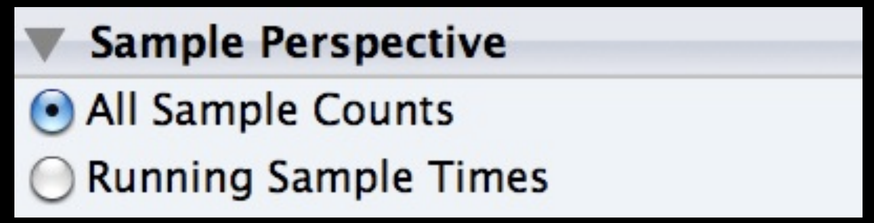
More details

▼ Sample Perspective
- (•) All Sample Counts
- ( ) Running Sample Times

Samples	Library	Symbol
151		▼ Thread 0x5203 : Main Thread
151	Breadcrumbs	▼ main
151	UIKit	▼ UIApplicationsMain
151	GraphicsServices	▼ GSEventRunModal
102	libSystem.B.dylib	mach_msg_trap
26	Breadcrumbs	▼ -[RootViewController tableView:cellForRowAtIndexPath:]
11	Breadcrumbs	► -[BreadcrumbCell initWithFrame:reuseIdentifier:]
9	Breadcrumbs	► -[BreadcrumbEntry thumbnail]
6	Breadcrumbs	► -[RootViewController entryForIndexPath:]
8	libobjc.A.dylib	objc_msgSend
2	WebCore	► -[NSString(WebStringDrawing) _web_sizeWithFont:]
2	CoreGraphics	► CGContextDrawImage
...	...	...

Statistical Sampling

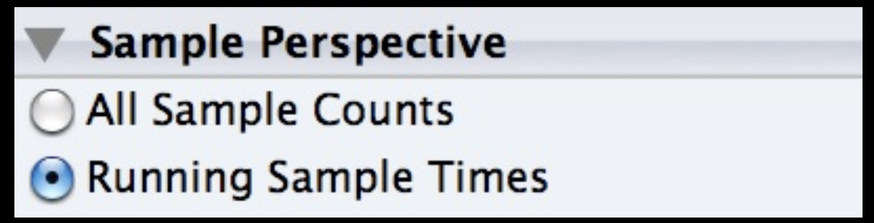
More details



Samples	Library	Symbol
151		▼ Thread 0x5203 : Main Thread
151	Breadcrumbs	▼ main
151	UIKit	▼ UIApplicationsMain
151	GraphicsServices	▼ GSEventRunModal
102	libSystem.B.dylib	mach_msg_trap
26	Breadcrumbs	▼ -[RootViewController tableView:cellForRowAtIndexPath:]
11	Breadcrumbs	► -[BreadcrumbCell initWithFrame:reuseIdentifier:]
9	Breadcrumbs	► -[BreadcrumbEntry thumbnail]
6	Breadcrumbs	► -[RootViewController entryForIndexPath:]
8	libobjc.A.dylib	objc_msgSend
2	WebCore	► -[NSString(WebStringDrawing) _web_sizeWithFont:]
2	CoreGraphics	► CGContextDrawImage
...	...	...

Statistical Sampling

More details



Running (ms)	Library	Symbol
380		▼ Thread 0x5203 : Main Thread
380	Breadcrumbs	▼ main
380	UIKit	▼ UIApplicationsMain
380	GraphicsServices	▼ GSEventRunModal
130	Breadcrumbs	▼ -[RootViewController tableView:cellForRowAtIndexPath:]
100	Breadcrumbs	► -[BreadcrumbCell initWithFrame:reuseIdentifier:]
30	Breadcrumbs	► -[RootViewController entryForIndexPath:]
0	Breadcrumbs	► -[BreadcrumbEntry thumbnail]
80	libobjc.A.dylib	objc_msgSend
50	WebCore	► -[NSString(WebStringDrawing) _web_sizeWithFont:]
20	CoreGraphics	► CGContextDrawImage
0	libSystem.B.dylib	mach_msg_trap
...	...	...

Statistical Sampling

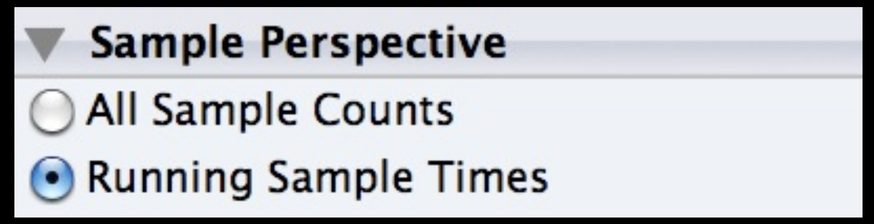
More details

▼ Sample Perspective
○ All Sample Counts
◉ Running Sample Times

Running (ms)	Library	Symbol
380		▼ Thread 0x5203 : Main Thread
380	Breadcrumbs	▼ main
380	UIKit	▼ UIApplicationsMain
380	GraphicsServices	▼ GSEventRunModal
130	Breadcrumbs	▼ -[RootViewController tableView:cellForRowAtIndexPath:]
100	Breadcrumbs	► -[BreadcrumbCell initWithFrame:reuseIdentifier:]
30	Breadcrumbs	► -[RootViewController entryForIndexPath:]
0	Breadcrumbs	► -[BreadcrumbEntry thumbnail]
80	libobjc.A.dylib	objc_msgSend
50	WebCore	► -[NSString(WebStringDrawing) _web_sizeWithFont:]
20	CoreGraphics	► CGContextDrawImage
0	libSystem.B.dylib	mach_msg_trap
...	...	...

Statistical Sampling

More details

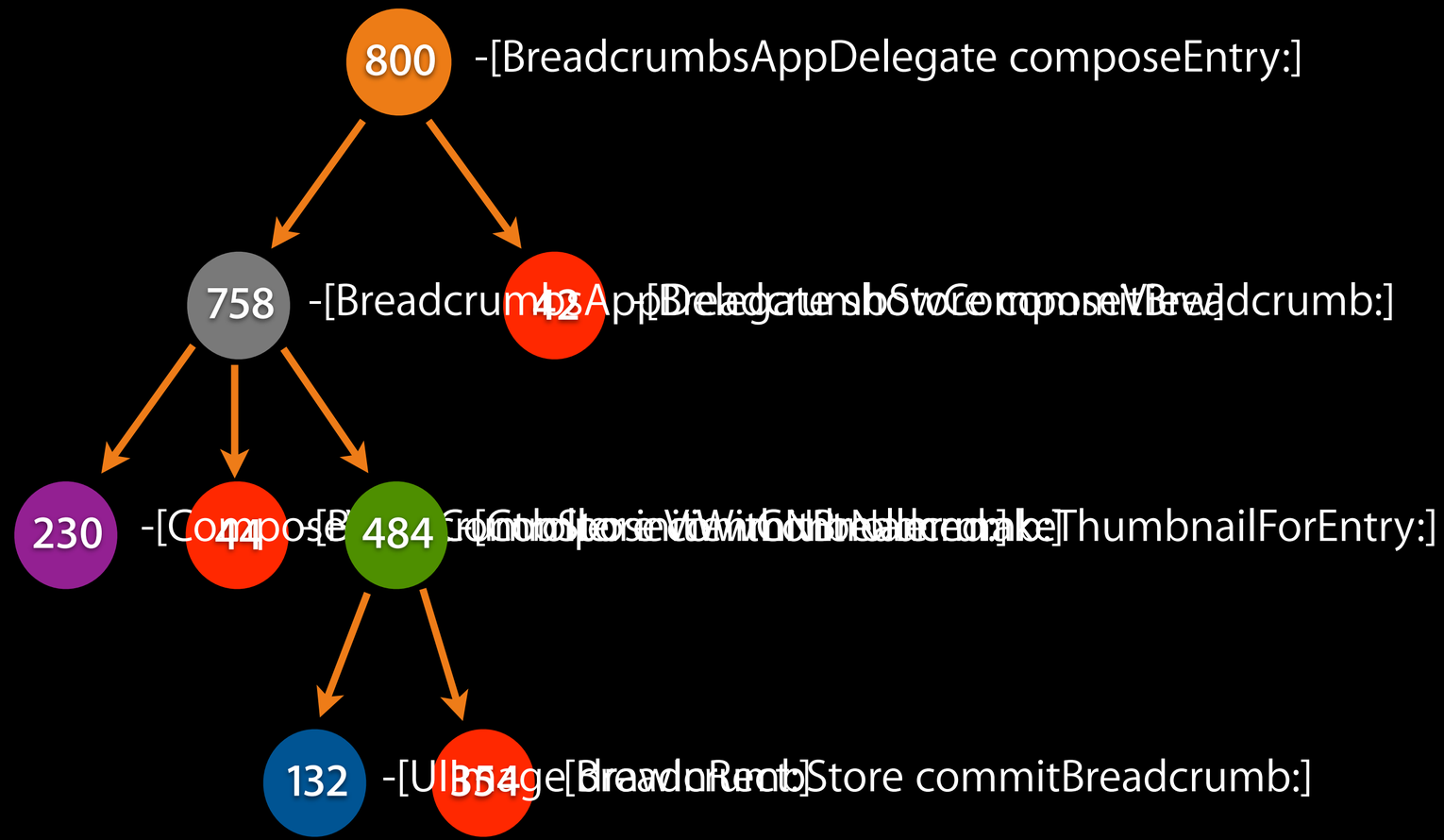


Running (ms)	Library	Symbol
380		▼ Thread 0x5203 : Main Thread
380	Breadcrumbs	▼ main
380	UIKit	▼ UIApplicationsMain
380	GraphicsServices	▼ GSEventRunModal
130	Breadcrumbs	▼ -[RootViewController tableView:cellForRowAtIndexPath:]
100	Breadcrumbs	► -[BreadcrumbCell initWithFrame:reuseIdentifier:]
30	Breadcrumbs	► -[RootViewController entryForIndexPath:]
0	Breadcrumbs	► -[BreadcrumbEntry thumbnail]
80	libobjc.A.dylib	objc_msgSend
50	WebCore	► -[NSString(WebStringDrawing) _web_sizeWithFont:]
20	CoreGraphics	► CGContextDrawImage
0	libSystem.B.dylib	mach_msg_trap
...	...	...

Statistical Sampling

More details

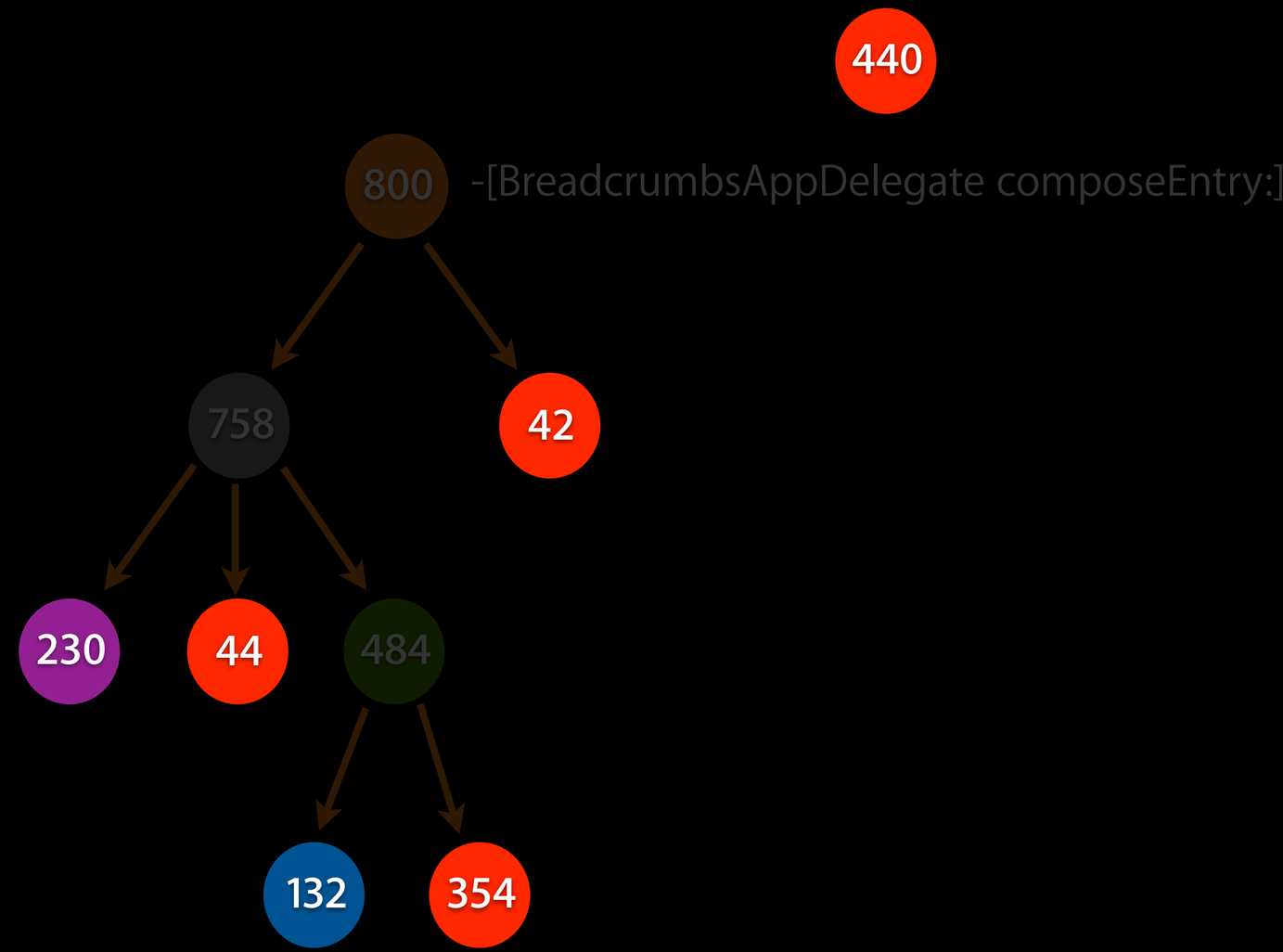
- Call Tree inversion
  - Off – Identify costs



Statistical Sampling

More details

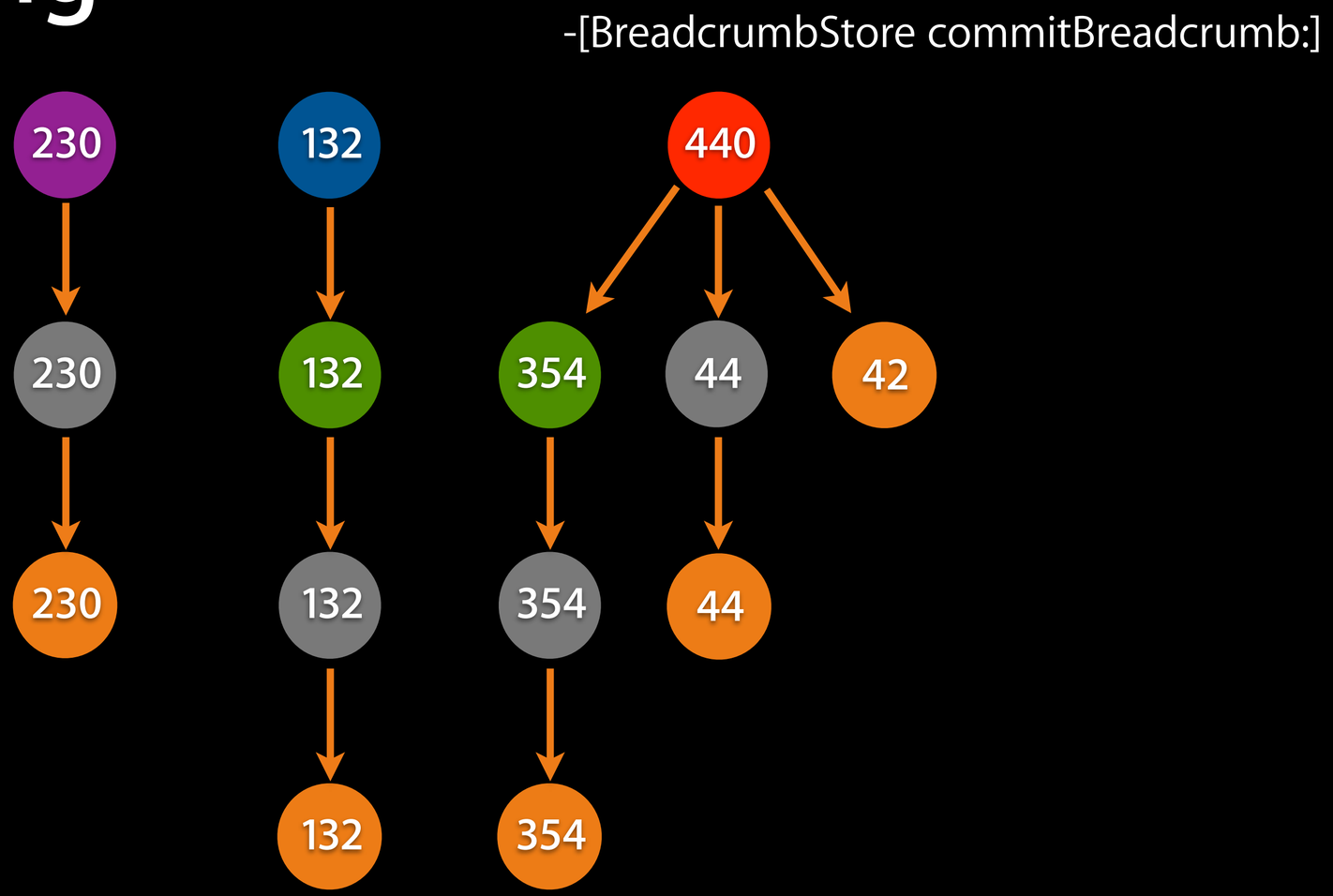
- Call Tree inversion
  - Off – Identify costs
  - On – Hotspots



Statistical Sampling

More details

- Call Tree inversion
  - Off – Identify costs
  - On – Hotspots



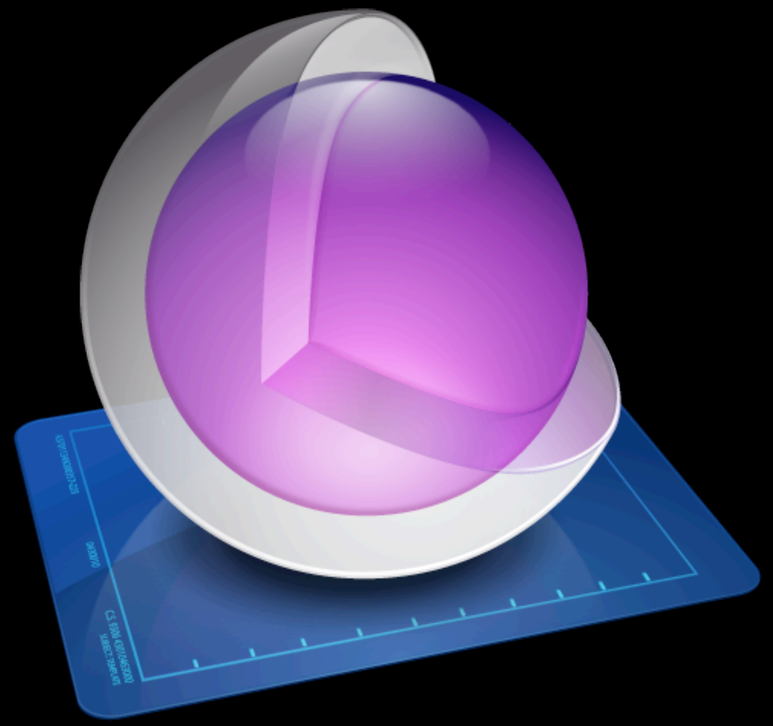
I/O Tools

I/O Tools

I/O Activity Instrument

- Records I/O events
- I/O takes a long time
- Sand-boxed apps
- Like ‘File Activity’

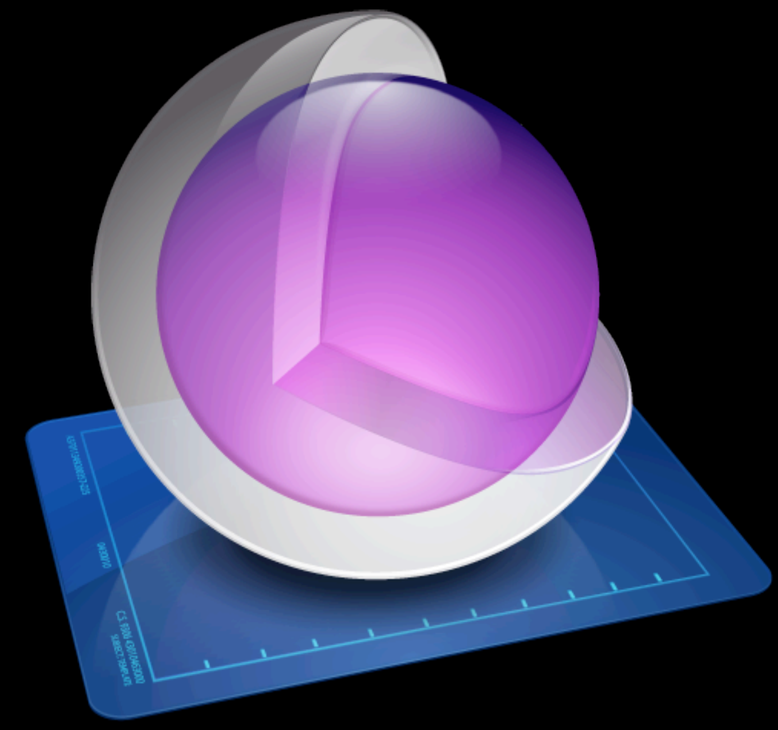
Drawing Tools



Drawing Tools

Core Animation

- Graphical annotations
  - Blending/opacity
  - Flash updates
  - Copied images
- Tracks overall framerate



Memory

Effective management

Memory Tools

Memory Tools

Object Alloc

- Tracks ‘malloc’ type memory
  - Objective-C, C, C++
- Malloc/Free/Retain/Release/Autorelease
- Statistics
- Call Trees
- Overhead

Memory Tools

Memory Tools

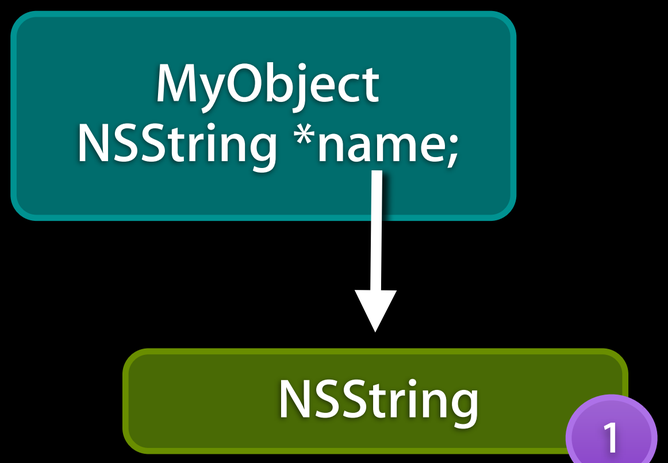
Leaks

- Static memory analysis
- Unreferenced memory blocks
- Works alongside ObjectAlloc

Memory Tools

Leaks

- Static memory analysis
- Unreferenced memory blocks
- Works alongside ObjectAlloc

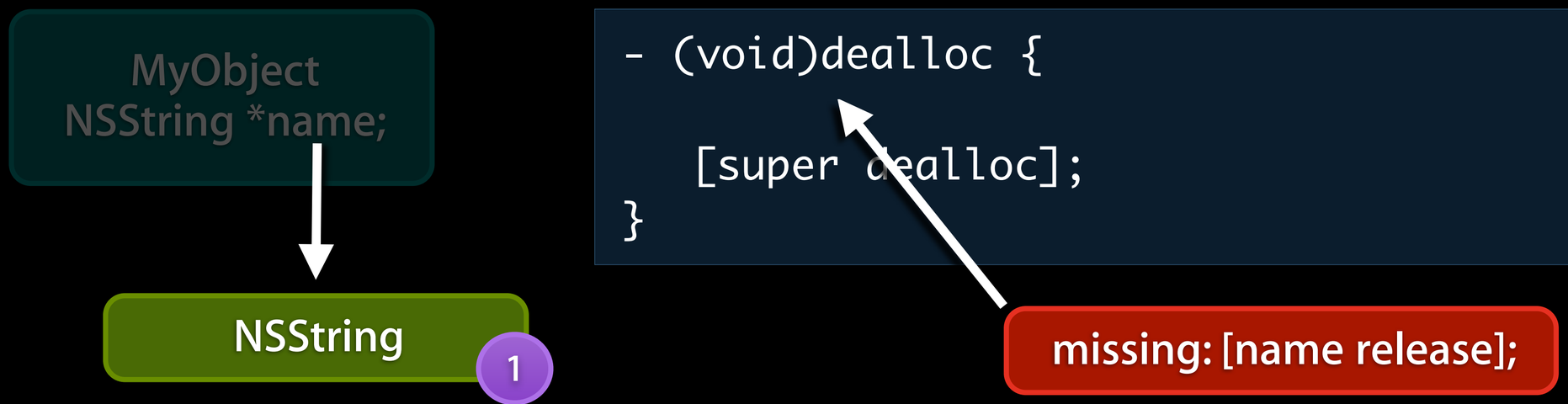



```
- (id)init {
    if (self = [super init]) {
        name = [[NSString alloc] initWithFormat:...];
    }
    return self;
}
```

Memory Tools

Leaks

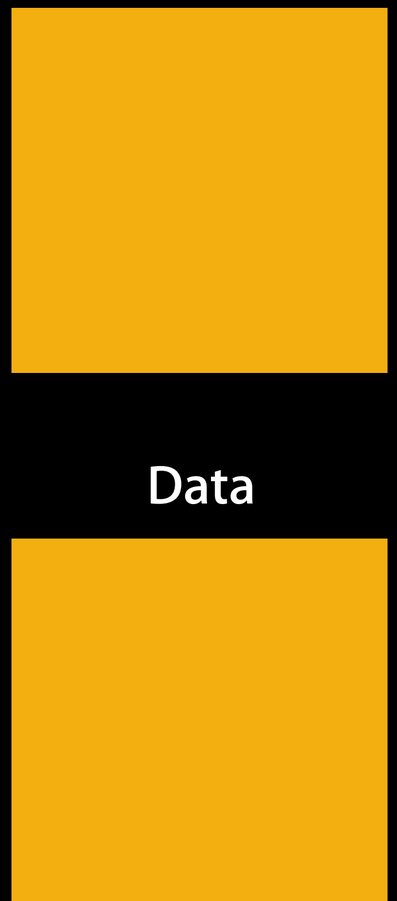
- Static memory analysis
- Unreferenced memory blocks
- Works alongside ObjectAlloc



Memory Tools

Leaks

Stacks

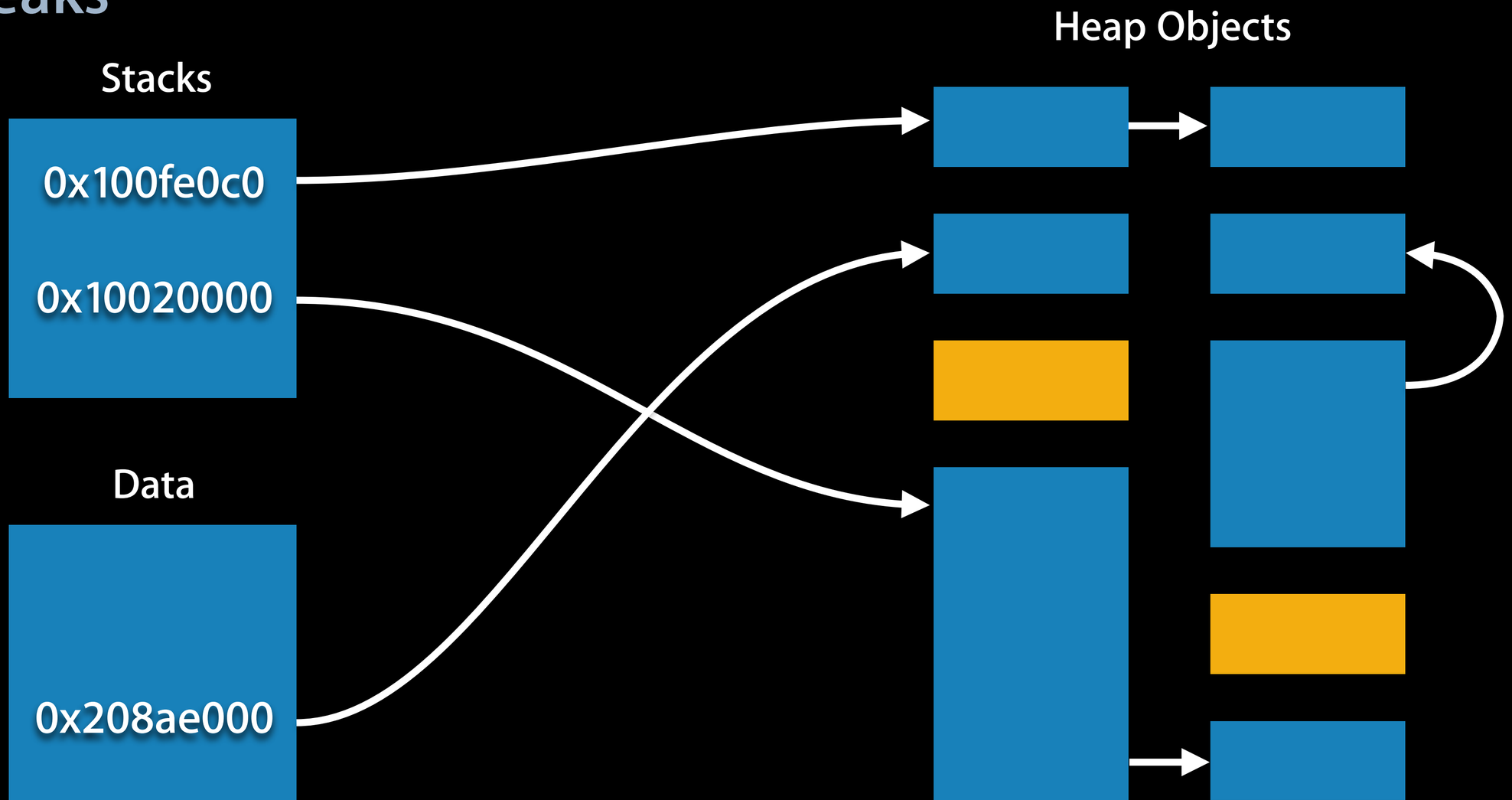


Data

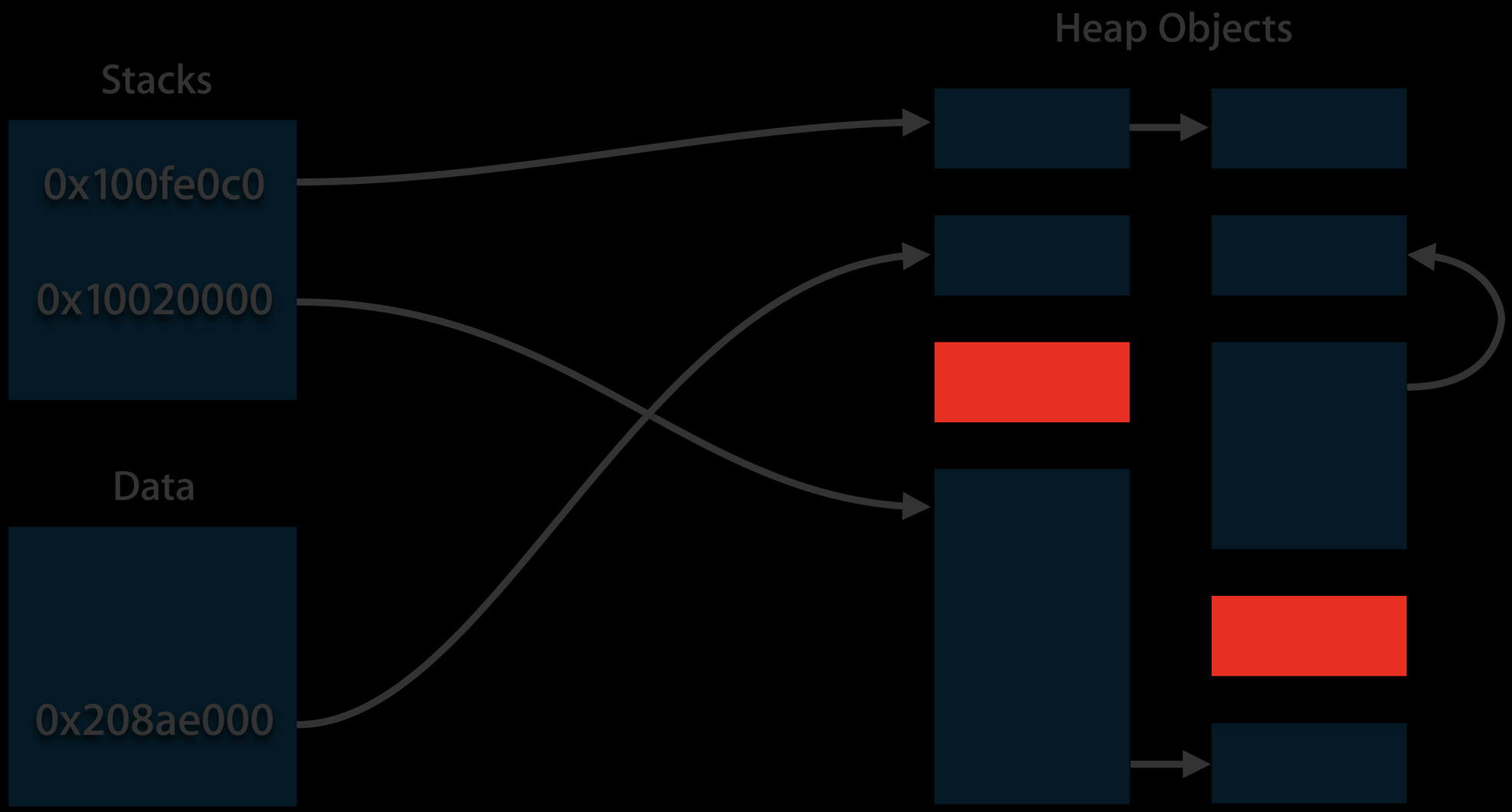
Heap Objects

Memory Tools

Leaks



Memory Tools



Leaks and Memory Growth Demo

Memory Tools

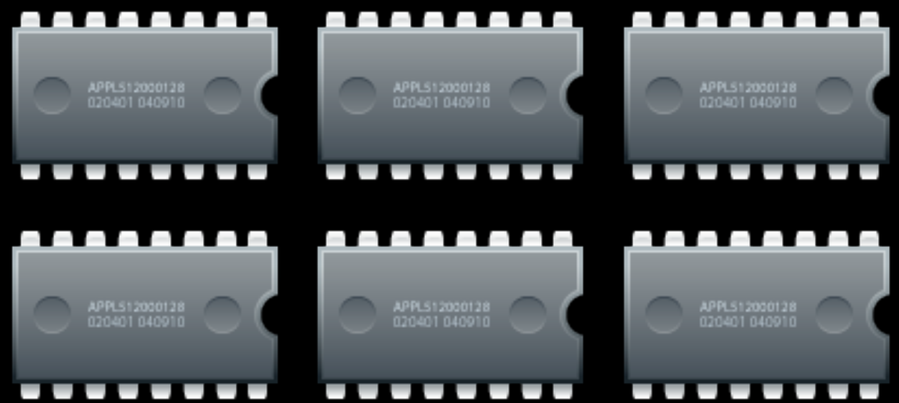
Leaks

- Conservative
  - Avoids false positives
  - Doesn't catch *everything*

Memory

Growth over time

- Be conscious of memory usage
- Actively investigate
- Prepare for low memory warnings



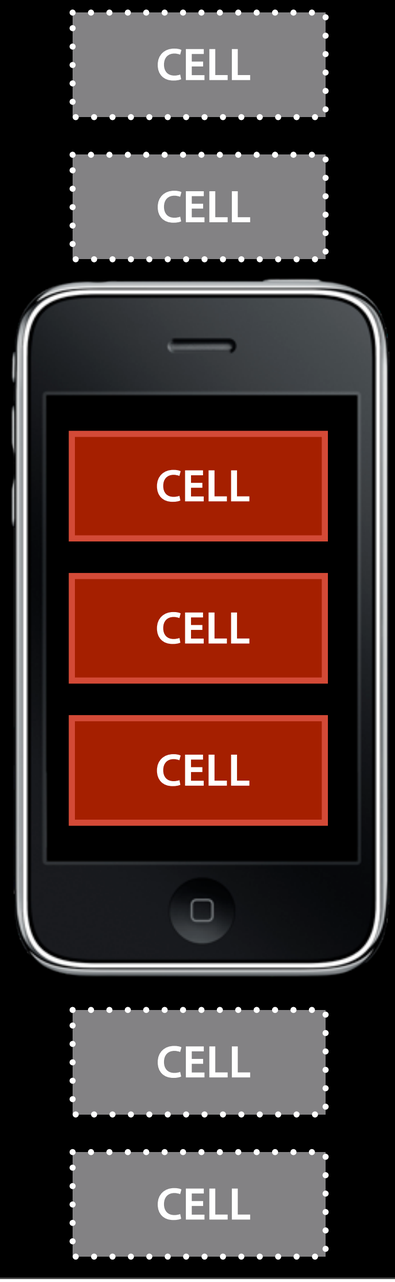
Memory Management Programming Guide for Cocoa

http://developer.apple.com/iphone/library/documentation/Cocoa/Conceptual/MemoryMgmt/

iPhone Specific

Cell Reuse

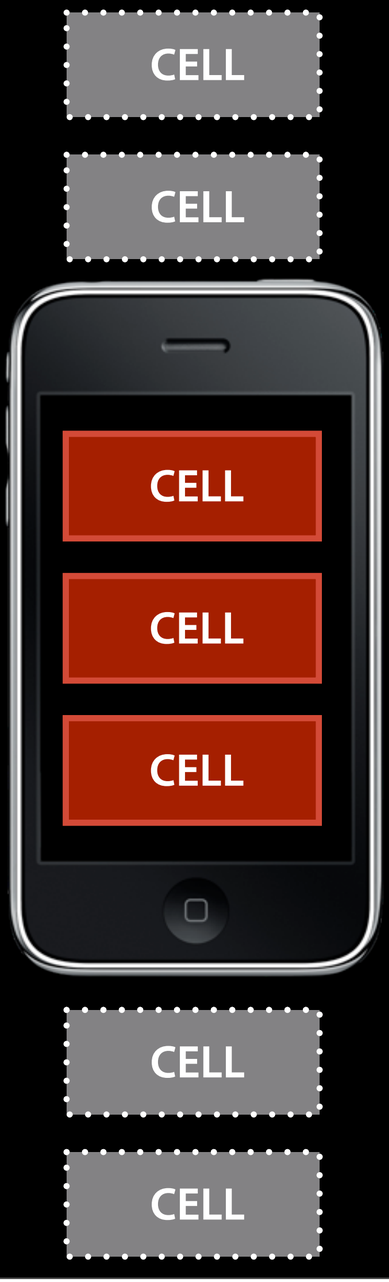
Before



iPhone Specific

Cell Reuse

After



Cell Reuse Demo

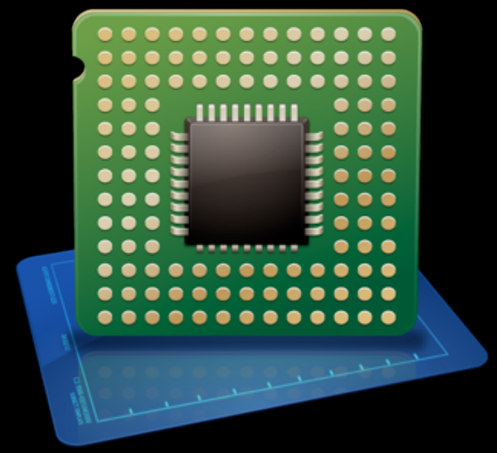
Reacting

Crashes, hangs, and getting killed

Be Responsive!

And if your app isn't…

- Users notice
- Your app gets terminated



- Don't block the main thread!
  - Deadlocks, network, infinite loops, etc.
  - Use Sampler to track down
  - Profile the device

Use Memory Wisely

Otherwise…

- System has to protect itself
  - App gets terminated!

- Fix leaks
- Reduce memory growth
- Respond to memory warnings
- Simulator convenience

```
Incident Identifier: 35B90F49-8199-42AD-BBE3-17818EB3D3D7
CrashReporter Key:   ca7b5d17293350bae6156251f26ab0f725142480
Process:         Breadcrumbs [112]
Path:            /var/mobile/Applications/17D54835-C6D1-481A-9356-9C57E9C064ED/Breadcrumbs.app/Breadcrumbs
Identifier:      Breadcrumbs
Version:         ??? (???)
Code Type:       ARM (Native)
Parent Process:  launchd [1]

Date/Time:       2009-07-11 00:00:00.00 -0700
OS Version:      iPhone OS 3.0
UDID:            21881ac512d48000a23092586cb60f9527e8bd39
Report Version:  104

Exception Type:  EXC_BAD_ACCESS (SIGBUS)
Exception Codes: KERN_PROTECTION_FAILURE at 0x00000000
Crashed Thread:  0

Thread 0 Crashed:
0   libobjc.A.dylib               0x30011944 objc_msgSend + 24
1   Breadcrumbs                   0x0000349e -[ComposeViewController dealloc] (ComposeViewController.m:90)
2   CoreFoundation                0x30236b72 -[NSObject release] + 28
3   CoreFoundation                0x30235f1e CFRelease + 98
4   CoreFoundation                0x302043b6 __CFTypeCollectionRelease + 30
5   CoreFoundation                0x30205afc __CFDictionaryDeallocate + 132
6   CoreFoundation                0x3023601a _CFRelease + 162
7   CoreFoundation                0x30235ef8 CFRelease + 60
8   Foundation                    0x30547f40 -[NSCFDictionary release] + 2
9   Foundation                    0x30548038 NSPopAutoreleasePool + 238
10  QuartzCore                    0x3293a4d4 run_animation_callbacks(double, void*) + 600
11  QuartzCore                    0x3293a23c CA::timer_callback(__CFRunLoopTimer*, void*) + 156
12  CoreFoundation                0x30254a08 CFRunLoopRunSpecific + 2192
13  CoreFoundation                0x30254164 CFRunLoopRunInMode + 44
14  GraphicsServices              0x3204529c GSEventRunModal + 188
15  UIKit                         0x308f0374 -[UIApplication _run] + 552
16  UIKit                         0x308eea8c UIApplicationMain + 960
17  Breadcrumbs                   0x000020b6 main (main.m:14)
18  Breadcrumbs                   0x0000202c start + 44

Thread 1:
0   libSystem.B.dylib             0x31d47158 mach_msg_trap + 20
1   libSystem.B.dylib             0x31d49ed8 mach_msg + 60
[...]
```

Exception Type: EXC_BAD_ACCESS (SIGBUS)
Exception Codes: KERN_PROTECTION_FAILURE at 0x00000000
Crashed Thread: 0

Thread 0 Crashed:

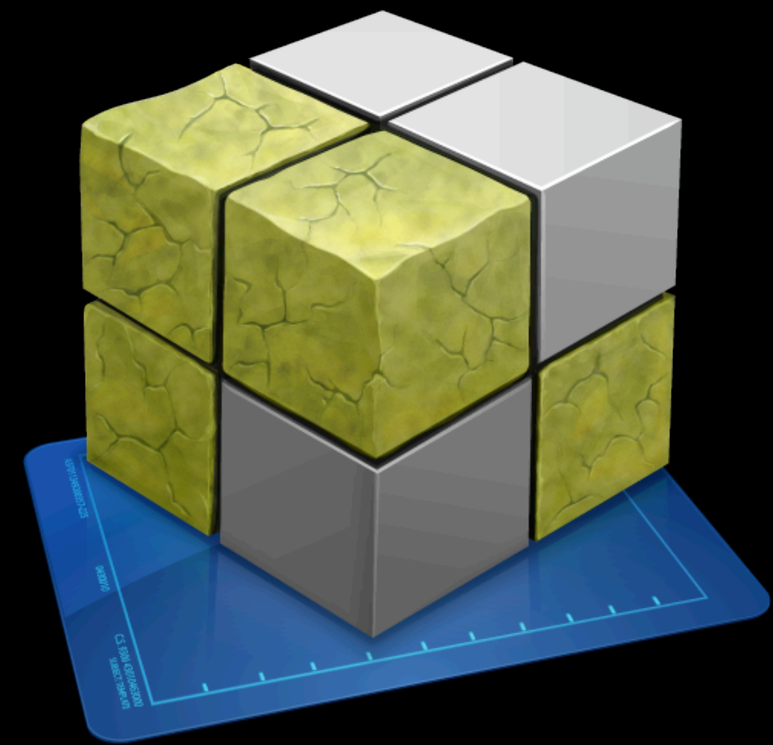
0	libobjc.A.dylib	0x30011944 objc_msgSend + 24
1	Breadcrumbs	0x0000349e -[ComposeViewController dealloc]
2	CoreFoundation	0x30236b72 -[NSObject release] + 28
3	CoreFoundation	0x30235f1e CFRelease + 98
4	CoreFoundation	0x302043b6 __CFTypeCollectionRelease + 30
5	CoreFoundation	0x30205afc __CFDictionaryDeallocate + 132
6	CoreFoundation	0x3023601a _CFRelease + 162
7	CoreFoundation	0x30235ef8 CFRelease + 60
8	Foundation	0x30547f40 -[NSCFDictionary release] + 2
9	Foundation	0x30548038 NSPopAutoreleasePool + 238
10	QuartzCore	0x3293a4d4 run_animation_callbacks(double, void*) + 600

Thread 1:
0 libSystem.B.dylib 0x31d47158 mach_msg_trap + 20
1 libSystem.B.dylib 0x31d49ed8 mach_msg + 60
[...]

Zombies

Over-released objects

```
[[NSString alloc] initWithFormat:...];
```



NSString

1

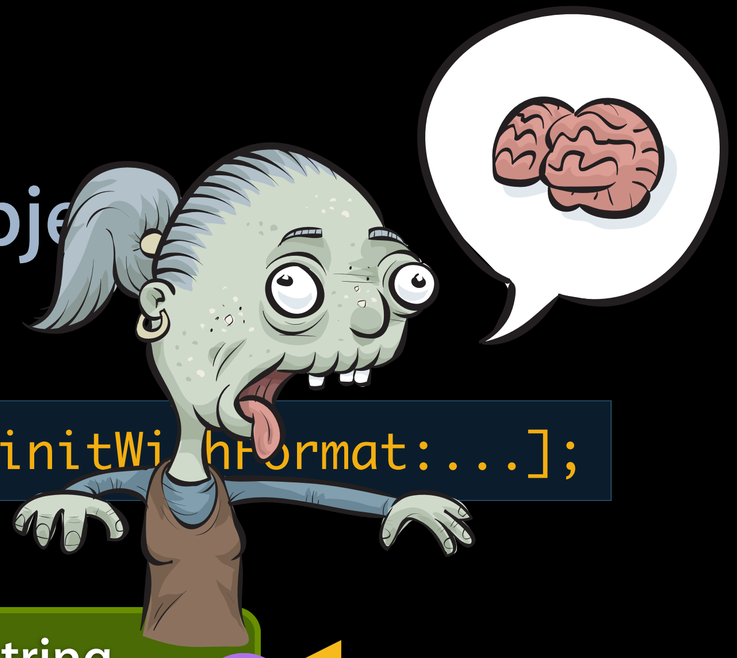
Crash

```
[string release];
```

```
[string stringByAppendingFormat:...];
```

Zombies

Over-released obje

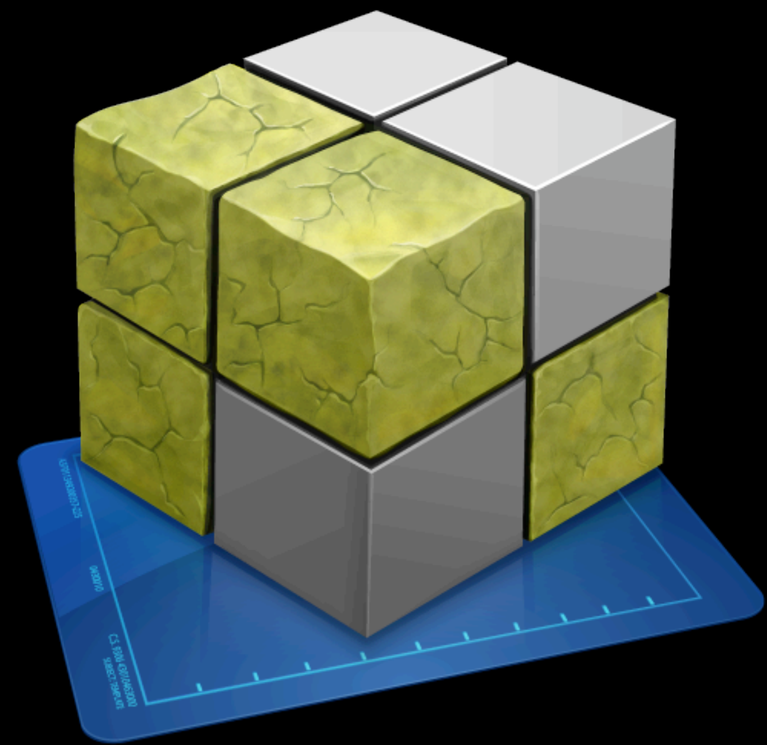


```
[[NSString alloc] initWi hrormat:...];
```

NSString 1

```
[string release];
```

```
[string stringByAppendingFormat:...];
```



BRAINS!

Zombies Demo

Summary

- Use instruments for iPhone
- Know simulator vs. device
- Be proactive about performance

More Information

Michael Jurewitz
Developer Tools Evangelist
jurewitz@apple.com

Instruments Documentation

Instruments User Guide (Xcode documentation)

Related Sessions

Beer Bash	Buena Vista Gardens Thursday 6:30PM

Related Labs

iPhone Performance Lab	iPhone Lab B Friday 9:00AM
iPhone Performance Optimization with Instruments Lab	iPhone Lab B Friday 2:00PM
Xcode Tools Lab	OS X Lab A Friday 9:00AM

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09